Laplace's Daemon

José Félix Costa

Departamento de Matemática, Instituto Superior Técnico

CMAF – Centro de Matemática e Aplicações Fundamentais da Faculdade de Ciências

CFCUL – Centro de Filosofia das Ciências da Universidade de Lisboa

Newton-Bentley's Correspondence

Introduction:
Newton-Bentley's correspondence

Newton-Bentley's correspondence led Newton to abandon to Stoic Cosmos of a finite distribution of matter in infinite space and to adopt the Atomist Universe in which matter is distributed throughout infinite space.

Introduction:
Newton-Bentley's correspondence

Newton-Bentley's correspondence led Newton to abandon to Stoic Cosmos of a finite distribution of matter in infinite space and to adopt the Atomist Universe in which matter is distributed throughout infinite space.

Letter 1

If the distribution of matter were finite, then the matter on the outside of this space would by its gravity tend toward the matter on the inside, and by consequence, fall down into the middle of the whole space, and there compose one great spherical mass... But if the matter was evenly diffused through an infinite space, it would never convene into one mass but some of it into one mass and some of it into another so as to make an infinite number of great masses scattered at great distances from one to another throughout all of infinite space. And thus might the Sun and fixed stars be formed.

Letter 1

If the distribution of matter were finite, then the matter on the outside of this space would by its gravity tend toward the matter on the inside, and by consequence, fall down into the middle of the whole space, and there compose one great spherical mass... But if the matter was evenly diffused through an infinite space, it would never convene into one mass but some of it into one mass and some of it into another so as to make an infinite number of great masses scattered at great distances from one to another throughout all of infinite space. And thus might the Sun and fixed stars be formed.

Letter 2

Newton had fully agreed with Bentley that gravity meant providence had created a universe of great precision.

Letter 3

The hypothesis of deriving the frame of the world by mechanical principles from matter evenly spread through the heavens being inconsistent with my system, I had considered it very little before your letters put me upon it, and therefore trouble you with a line or two more about it...

Letter 2

Newton had fully agreed with Bentley that gravity meant providence had created a universe of great precision.

Letter 3

The hypothesis of deriving the frame of the world by mechanical principles from matter evenly spread through the heavens being inconsistent with my system, I had considered it very little before your letters put me upon it, and therefore trouble you with a line or two more about it...

Letter 4

Newton elaborated earlier arguments that a divine power was essential in the design of initial conditions.

... this frame of things could not always subsist without a divine power to conserve it.

Letter 4

Newton elaborated earlier arguments that a divine power was essential in the design of initial conditions.

... this frame of things could not always subsist without a divine power to conserve it.

Newton's obsession at an old age

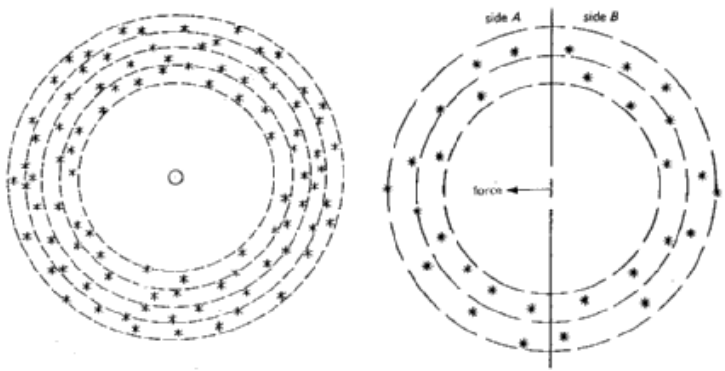


Figure: *Sensorium Dei* in Newton's metaphysics.

Given the initial conditions with finite precision, a human being can not:

- Ω Calculate with the same accuracy the state of the universe at a given time t, and can not prove properties of the universe, such like *its trajectory will not cross a given finite region of space*.

Given the initial conditions with finite precision, a human being can not:

- Ω Calculate with the same accuracy the state of the universe at a given time t, and can not prove properties of the universe, such like *its trajectory will not cross a given finite region of space*.

A "deus ex machina" can not:

Ω_1 Know the initial conditions of the universe with infinite precision;

Ω_2 Observe the state of the universe with infinite precision.

A "deus ex machina" can not:

Ω_1 Know the initial conditions of the universe with infinite precision;

Ω_2 Observe the state of the universe with infinite precision.

A "deus ex machina" can not:

Ω_1 Know the initial conditions of the universe with infinite precision;

Ω_2 Observe the state of the universe with infinite precision.

Given the initial conditions with infinite precision, a "deus ex machina" can either... or...:

Ω_1 Calculate the state of the universe at a given time t and prove properties of the universe, such like *its trajectory will not cross a given finite region of space*.

Ω_2 Calculate the state of the universe at a given time t, but it can not prove properties of the universe such like *its trajectory will not cross a given finite region of space*.

Given the initial conditions with infinite precision, a "deus ex machina" can either... or...:

Ω_1 Calculate the state of the universe at a given time t and prove properties of the universe, such like *its trajectory will not cross a given finite region of space*.

Ω_2 Calculate the state of the universe at a given time t, but it can not prove properties of the universe such like *its trajectory will not cross a given finite region of space*.

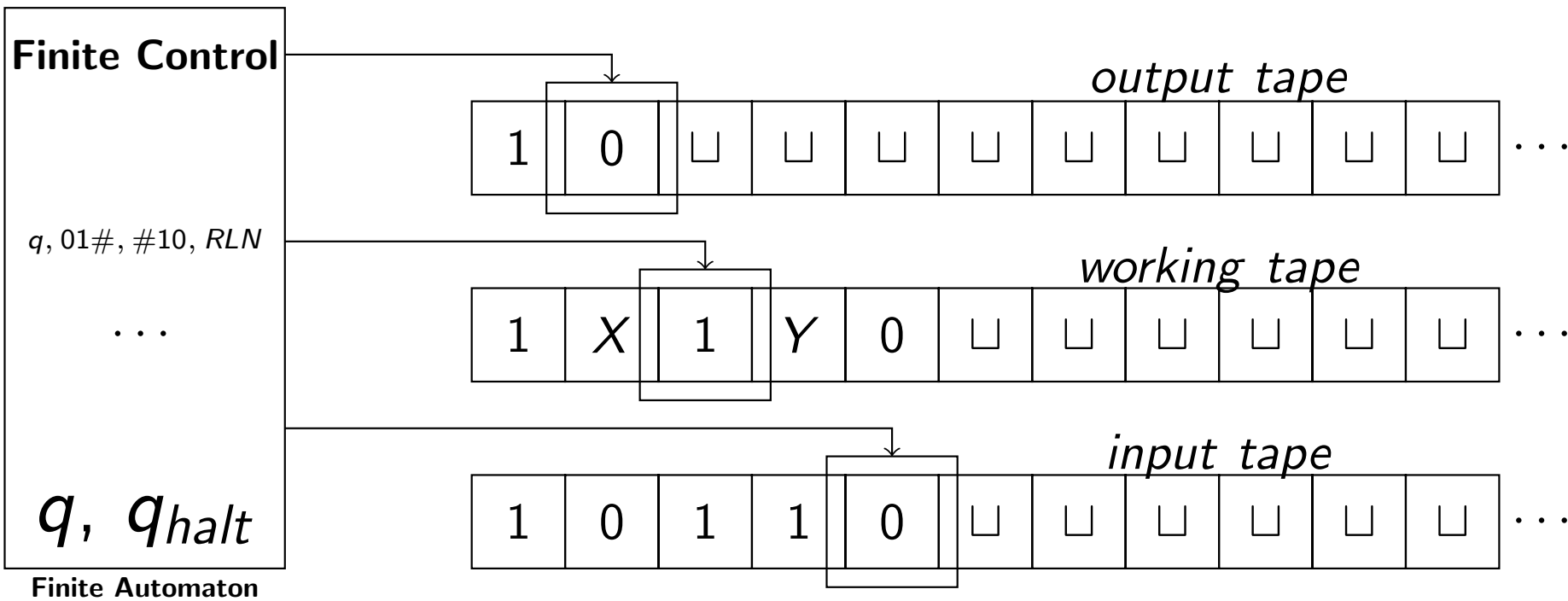
Given the initial conditions with infinite precision, a "deus ex machina" can either... or...:

Ω_1 Calculate the state of the universe at a given time t and prove properties of the universe, such like *its trajectory will not cross a given finite region of space*.

Ω_2 Calculate the state of the universe at a given time t, but it can not prove properties of the universe such like *its trajectory will not cross a given finite region of space*.

The Turing Machine

The Turing machine



The Turing machine

- A Turing machine can be seen as a formal definition of algorithm; more abstract linguistic constructs can be implemented on top of the Turing machine. Thus, for algorithm specification purpose, we can use a high-level equivalent programming language.
- **Turing machine with $k > 2$ tapes:**
  $\delta : Q \times \Gamma^{k-1} \to Q \times \Gamma^{k-1} \times \{L, N, R\}^k$
- **Example**

  input n;
  while $n \neq 1$ do
  if $even(n)$ then $n := n/2$ else $n = 3n + 1$
  end while

The Turing machine

- A Turing machine can be seen as a formal definition of algorithm; more abstract linguistic constructs can be implemented on top of the Turing machine. Thus, for algorithm specification purpose, we can use a high-level equivalent programming language.
- **Turing machine with $k > 2$ tapes:**
  $\delta : Q \times \Gamma^{k-1} \to Q \times \Gamma^{k-1} \times \{L, N, R\}^k$
- **Example**

  input n;
  while $n \neq 1$ do
  if $even(n)$ then $n := n/2$ else $n = 3n + 1$
  end while

The Turing machine

- A Turing machine can be seen as a formal definition of algorithm; more abstract linguistic constructs can be implemented on top of the Turing machine. Thus, for algorithm specification purpose, we can use a high-level equivalent programming language.
- **Turing machine with $k > 2$ tapes:**
  $\delta : Q \times \Gamma^{k-1} \rightarrow Q \times \Gamma^{k-1} \times \{L, N, R\}^k$
- **Example**

  input n;
  while $n \neq 1$ do
  if $even(n)$ then $n := n/2$ else $n = 3n + 1$
  end while

The Turing machine

- A Turing machine can be seen as a formal definition of algorithm; more abstract linguistic constructs can be implemented on top of the Turing machine. Thus, for algorithm specification purpose, we can use a high-level equivalent programming language.
- **Turing machine with $k > 2$ tapes:**
  $\delta : Q \times \Gamma^{k-1} \rightarrow Q \times \Gamma^{k-1} \times \{L, N, R\}^k$
- **Example**

  input n;
  while $n \neq 1$ do
  if $even(n)$ then $n := n/2$ else $n = 3n + 1$
  end while

Semantics of algorithm

- **Computable function** Each Turing machine $\mathcal{M}$ (with an output tape) computes a function $f_{\mathcal{M}} : \{0,1\}^\star \to \{0,1\}^\star$ (in other words, a function $f_{\mathcal{M}} : \mathbb{N} \to \mathbb{N}$); if the machine does not halt on an input x, then it is said that $f_{\mathcal{M}}$ is not defined at x.

- **Bijection between $\mathbb{N}$ and $\{0,1\}^\star$** Take a binary word, e.g., 0101, affix a leftmost 1 (10101), and read it in binary having subtracted 1 (20).
  Take any number in decimal, e.g., 20, add 1, write it in binary (10101), remove the leftmost 1, and read the result as a binary word (0101).

Semantics of algorithm

- **Computable function** Each Turing machine $\mathcal{M}$ (with an output tape) computes a function $f_{\mathcal{M}} : \{0,1\}^\star \to \{0,1\}^\star$ (in other words, a function $f_{\mathcal{M}} : \mathbb{N} \to \mathbb{N}$); if the machine does not halt on an input x, then it is said that $f_{\mathcal{M}}$ is not defined at x.

- **Bijection between $\mathbb{N}$ and $\{0,1\}^\star$** Take a binary word, e.g., 0101, affix a leftmost 1 (10101), and read it in binary having subtracted 1 (20).
  Take any number in decimal, e.g., 20, add 1, write it in binary (10101), remove the leftmost 1, and read the result as a binary word (0101).

Semantics of algorithm

- **Computable function** Each Turing machine $\mathcal{M}$ (with an output tape) computes a function $f_{\mathcal{M}} : \{0,1\}^\star \to \{0,1\}^\star$ (in other words, a function $f_{\mathcal{M}} : \mathbb{N} \to \mathbb{N}$); if the machine does not halt on an input x, then it is said that $f_{\mathcal{M}}$ is not defined at x.

- **Bijection between $\mathbb{N}$ and $\{0,1\}^\star$** Take a binary word, e.g., 0101, affix a leftmost 1 (10101), and read it in binary having subtracted 1 (20).
  Take any number in decimal, e.g., 20, add 1, write it in binary (10101), remove the leftmost 1, and read the result as a binary word (0101).

Theorem

There is a countable infinite number of computable functions, but an uncountable number of non-computable functions. (I.e., most of the functions $f : \{0,1\}^\star \to \{0,1\}^\star$ are non-algorithmic.)

Theorem (Universal Turing machine)

There exists a Turing machine $\mathcal{U}$ that receives as input $\langle \mathcal{M}, x \rangle$, the binary code of a Turing machine $\mathcal{M}$ and a binary word x, such that, for every such $\mathcal{M}$ and for every such x, it simulates $\mathcal{M}$ on input x:

$$\mathcal{U}(\langle \mathcal{M}, x \rangle) \equiv \mathcal{M}(x) \ .$$

Theorem

There is a countable infinite number of computable functions, but an uncountable number of non-computable functions. (I.e., most of the functions $f : \{0,1\}^\star \to \{0,1\}^\star$ are non-algorithmic.)

Theorem (Universal Turing machine)

There exists a Turing machine $\mathcal{U}$ that receives as input $\langle \mathcal{M}, x \rangle$, the binary code of a Turing machine $\mathcal{M}$ and a binary word x, such that, for every such $\mathcal{M}$ and for every such x, it simulates $\mathcal{M}$ on input x:

$$\mathcal{U}(\langle \mathcal{M}, x \rangle) \equiv \mathcal{M}(x) \ .$$

Theorem

The Turing machine can not compute functions with arbitrary growing rate. There is a well-known established limit of growing rate for computable functions.

Theorem

The halting function, namely the function

$$f(\langle y, x\rangle) = \begin{cases} 1 & \text{if } \mathcal{M}_y \text{ halts on } x \\ 0 & \text{otherwise} \end{cases}$$

is not computable.

Theorem

The Turing machine can not compute functions with arbitrary growing rate. There is a well-known established limit of growing rate for computable functions.

Theorem

The halting function, namely the function

$$f(\langle y, x\rangle) = \begin{cases} 1 & \text{if } \mathcal{M}_y \text{ halts on } x \\ 0 & \text{otherwise} \end{cases}$$

is not computable.

Open Problem

Collatz function It is not known if the following Turing machine halts or not, for all natural inputs n:

input n;
while $n \neq 1$ do
    if $even(n)$ then $n := n/2$ else $n = 3n + 1$
end while

Example Sequences of numbers produced for different inputs:

Input 4 , 2, 1 HALT
Input 5 , 16, 8, 4, 2, 1 HALT
Input 7 , 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1 HALT

Open Problem

Collatz function It is not known if the following Turing machine halts or not, for all natural inputs n:

input n;
while $n \neq 1$ do
    if $even(n)$ then $n := n/2$ else $n = 3n + 1$
end while

Example Sequences of numbers produced for different inputs:

Input 4 , 2, 1 HALT
Input 5 , 16, 8, 4, 2, 1 HALT
Input 7 , 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1 HALT

Open Problem

Collatz function It is not known if the following Turing machine halts or not, for all natural inputs n:

input n;
while $n \neq 1$ do
    if $even(n)$ then $n := n/2$ else $n = 3n + 1$
end while

Example Sequences of numbers produced for different inputs:

Input 4 , 2, 1 HALT
Input 5 , 16, 8, 4, 2, 1 HALT
Input 7 , 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1 HALT

Definition

A set A (e.g., a subset of $\{0,1\}^\star$) is said to be decidable if the (characteristic) function (of A)

$$\xi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

is Turing computable.

A set A (e.g., a subset of $\{0,1\}^\star$) is said to be semidecidable if the (partial characteristic) function (of A)

$$\xi\!\!\!/_A(x) = \begin{cases} 1 & \text{if } x \in A \\ \bot & \text{if } x \notin A \end{cases}$$

is Turing computable.

Definition (Diophantine sets)

We say that a relation D is Diophantine if there exists a polynomial p with integer coefficients, such that,

$$\langle m_1, \ldots, m_k \rangle \in D \text{ iff } \exists x_1, \ldots, x_n \in \mathbb{N} \, [\, p(m_1, \ldots, m_k, x_1, \ldots, x_n) = 0 \,] \, .$$

Examples Composite numbers, divisible numbers, prime numbers, etc.:

$$\begin{aligned} x \in \mathit{Composites} \quad &\text{iff} \quad \exists y, z \in \mathbb{N} \, [\, yz + y + z - x + 1 = 0 \,] \\ x|y \quad &\text{iff} \quad \exists z \in \mathbb{N} \, [\, xz - y = 0 \,] \\ x|y \text{ and } x < y \quad &\text{iff} \quad \exists u, v \in \mathbb{N} \, [\, (xu - y)^2 + (y - x - v - 1)^2 = 0 \,] \end{aligned}$$

Definition (Diophantine sets)

We say that a relation D is Diophantine if there exists a polynomial p with integer coefficients, such that,

$$\langle m_1, \ldots, m_k \rangle \in D \text{ iff } \exists x_1, \ldots, x_n \in \mathbb{N} \, [\, p(m_1, \ldots, m_k, x_1, \ldots, x_n) = 0 \,] \, .$$

Examples Composite numbers, divisible numbers, prime numbers, etc.:

$$\begin{aligned} x \in \mathit{Composites} \quad &\text{iff} \quad \exists y, z \in \mathbb{N} \, [\, yz + y + z - x + 1 = 0 \,] \\ x|y \quad &\text{iff} \quad \exists z \in \mathbb{N} \, [\, xz - y = 0 \,] \\ x|y \text{ and } x < y \quad &\text{iff} \quad \exists u, v \in \mathbb{N} \, [\, (xu - y)^2 + (y - x - v - 1)^2 = 0 \,] \end{aligned}$$

Diophantine sets

[Davis, Putnam, Robinson, Matiyasevich (1970)]

Theorem

A set is Diophantine if and only if it is semidecidable.

Theorem

A set is Diophantine if and only if its partial characteristic is Turing computable.

Diophantine sets

[Davis, Putnam, Robinson, Matiyasevich (1970)]

Theorem

A set is Diophantine if and only if it is semidecidable.

Theorem

A set is Diophantine if and only if its partial characteristic is Turing computable.

Diophantine sets

[Davis, Putnam, Robinson, Matiyasevich (1970)]

Theorem

A set is Diophantine if and only if it is semidecidable.

Theorem

A set is Diophantine if and only if its partial characteristic is Turing computable.

Sturm's Algorithm

Theorem

There is an algorithm to compute the number and an interval containing the zeros of a polynomial of integer coefficients.

Comment For the physicist, the undecidability of Hilbert's tenth problem may appear intriguing.

Sturm's Algorithm

Theorem

There is an algorithm to compute the number and an interval containing the zeros of a polynomial of integer coefficients.

Comment For the physicist, the undecidability of Hilbert's tenth problem may appear intriguing.

Sturm's Algorithm

Theorem

There is an algorithm to compute the number and an interval containing the zeros of a polynomial of integer coefficients.

Comment For the physicist, the undecidability of Hilbert's tenth problem may appear intriguing.

Church-Turing Thesis

Postulate

No reasonable definition of algorithm produces more computable functions than the Turing machine.

Evidences

- The Turing machine mimics the activity of a computor.
- No one ever specified an intuitively algorithmic function that escapes to the computability criterion of the Turing machine.
- Different concepts of abstract device produce exactly the same class of functions.

Church-Turing Thesis

Postulate

No reasonable definition of algorithm produces more computable functions than the Turing machine.

Evidences

- The Turing machine mimics the activity of a computor.
- No one ever specified an intuitively algorithmic function that escapes to the computability criterion of the Turing machine.
- Different concepts of abstract device produce exactly the same class of functions.

Church-Turing Thesis

Postulate

No reasonable definition of algorithm produces more computable functions than the Turing machine.

Evidences

- The Turing machine mimics the activity of a computor.
- No one ever specified an intuitively algorithmic function that escapes to the computability criterion of the Turing machine.
- Different concepts of abstract device produce exactly the same class of functions.

Church-Turing Thesis

Postulate

No reasonable definition of algorithm produces more computable functions than the Turing machine.

Evidences

- The Turing machine mimics the activity of a computor.
- No one ever specified an intuitively algorithmic function that escapes to the computability criterion of the Turing machine.
- Different concepts of abstract device produce exactly the same class of functions.

Church-Turing Thesis

Postulate

No reasonable definition of algorithm produces more computable functions than the Turing machine.

Evidences

- The Turing machine mimics the activity of a computor.
- No one ever specified an intuitively algorithmic function that escapes to the computability criterion of the Turing machine.
- Different concepts of abstract device produce exactly the same class of functions.

Church-Turing Thesis

Postulate

No reasonable definition of algorithm produces more computable functions than the Turing machine.

Evidences

- The Turing machine mimics the activity of a computor.
- No one ever specified an intuitively algorithmic function that escapes to the computability criterion of the Turing machine.
- Different concepts of abstract device produce exactly the same class of functions.

Bounded Resources

Bounded Resources — What are they, really?

Space and time constructibility: Let x be an input of length $n = |x|$.

Definition

A function $f : \mathbb{N} \to \mathbb{N}$ is said to be time constructible if and only if there exists a Turing machine $\mathcal{M}$ such that, for every input x, the machine halts exactly in $f(|x|)$ steps.

A function $f : \mathbb{N} \to \mathbb{N}$ is said to be space constructible if and only if there exists a Turing machine $\mathcal{M}$ such that, for every input x, the machine halts exactly in a configuration in which exactly $f(|x|)$ tape cells are non-blanc and no other workspace has been used during the computation.

Bounded Resources — What are they, really?

Space and time constructibility: Let x be an input of length $n = |x|$.

Definition

A function $f : \mathbb{N} \to \mathbb{N}$ is said to be time constructible if and only if there exists a Turing machine $\mathcal{M}$ such that, for every input x, the machine halts exactly in $f(|x|)$ steps.

A function $f : \mathbb{N} \to \mathbb{N}$ is said to be space constructible if and only if there exists a Turing machine $\mathcal{M}$ such that, for every input x, the machine halts exactly in a configuration in which exactly $f(|x|)$ tape cells are non-blanc and no other workspace has been used during the computation.

Bounded Resources — What are they, really?

Space and time constructibility: Let x be an input of length $n = |x|$.

Definition

A function $f : \mathbb{N} \to \mathbb{N}$ is said to be time constructible if and only if there exists a Turing machine $\mathcal{M}$ such that, for every input x, the machine halts exactly in $f(|x|)$ steps.

A function $f : \mathbb{N} \to \mathbb{N}$ is said to be space constructible if and only if there exists a Turing machine $\mathcal{M}$ such that, for every input x, the machine halts exactly in a configuration in which exactly $f(|x|)$ tape cells are non-blanc and no other workspace has been used during the computation.

Facts on constructibility

Theorem

Time and space constructible functions are computable functions. Not all computable functions are time or space constructible.

Functions with the expression n^k, 2^{kn}, 2^{n^k} are time-constructible.

Functions with the expression $\log n$, n^k, 2^{kn} are space-constructible.

Facts on constructibility

Theorem

Time and space constructible functions are computable functions.
Not all computable functions are time or space constructible.

Functions with the expression n^k, 2^{kn}, 2^{n^k} are time-constructible.

Functions with the expression $\log n$, n^k, 2^{kn} are space-constructible.

Facts on constructibility

Theorem

Time and space constructible functions are computable functions. Not all computable functions are time or space constructible.

Functions with the expression n^k, 2^{kn}, 2^{n^k} are time-constructible.

Functions with the expression $\log n$, n^k, 2^{kn} are space-constructible.

Facts on constructibility

Theorem

Time and space constructible functions are computable functions. Not all computable functions are time or space constructible.

Functions with the expression n^k, 2^{kn}, 2^{n^k} *are time-constructible.*

Functions with the expression $\log n$, n^k, 2^{kn} *are space-constructible.*

Machine in constructible space s:

$\mathcal{M}$ is a Turing machine;

$\mathcal{M}'$ is a witness of s;

$\mathcal{M}'; \mathcal{M}$ is a new machine $\widehat{\mathcal{M}}$ marked with s.

Just modify the finite control of $\mathcal{M}$ to incorporate the finite control of $\mathcal{M}'$; a mark is put at cell $s(|x|)$ and all the other cells are cleaned.

Example of a complexity class:

$$\begin{aligned} PSPACE \quad = \quad & \{\, A \subseteq \{0,1\}^* : \\ & A \text{ is decidable by } \mathcal{M}'; \mathcal{M}, \\ & \text{where } \mathcal{M} \text{ is a Turing machine and} \\ & \mathcal{M}' \text{ witnesses space } n^k \text{ for some } k \,\} \end{aligned}$$

Machine in constructible space s:

$\mathcal{M}$ is a Turing machine;

$\mathcal{M}'$ is a witness of s;

$\mathcal{M}'; \mathcal{M}$ is a new machine $\widehat{\mathcal{M}}$ marked with s.

Just modify the finite control of $\mathcal{M}$ to incorporate the finite control of $\mathcal{M}'$; a mark is put at cell $s(|x|)$ and all the other cells are cleaned.

Example of a complexity class:

$$\begin{aligned} PSPACE \quad = \quad & \{\, A \subseteq \{0,1\}^\star : \\ & A \text{ is decidable by } \mathcal{M}'; \mathcal{M}, \\ & \text{where } \mathcal{M} \text{ is a Turing machine and} \\ & \mathcal{M}' \text{ witnesses space } n^k \text{ for some } k \,\} \end{aligned}$$

Machine in constructible space s:

$\mathcal{M}$ is a Turing machine;

$\mathcal{M}'$ is a witness of s;

$\mathcal{M}'; \mathcal{M}$ is a new machine $\widehat{\mathcal{M}}$ marked with s.

Just modify the finite control of $\mathcal{M}$ to incorporate the finite control of $\mathcal{M}'$; a mark is put at cell $s(|x|)$ and all the other cells are cleaned.

Example of a complexity class:

$$\begin{aligned} PSPACE \quad = \quad & \{\, A \subseteq \{0,1\}^* : \\ & A \text{ is decidable by } \mathcal{M}'; \mathcal{M}, \\ & \text{where } \mathcal{M} \text{ is a Turing machine and} \\ & \mathcal{M}' \text{ witnesses space } n^k \text{ for some } k \,\} \end{aligned}$$

Machine in constructible space s:

$\mathcal{M}$ is a Turing machine;

$\mathcal{M}'$ is a witness of s;

$\mathcal{M}'; \mathcal{M}$ is a new machine $\widehat{\mathcal{M}}$ marked with s.

Just modify the finite control of $\mathcal{M}$ to incorporate the finite control of $\mathcal{M}'$; a mark is put at cell $s(|x|)$ and all the other cells are cleaned.

Example of a complexity class:

$$\begin{aligned} PSPACE \quad = \quad & \{\, A \subseteq \{0,1\}^* : \\ & A \text{ is decidable by } \mathcal{M}'; \mathcal{M}, \\ & \text{where } \mathcal{M} \text{ is a Turing machine and} \\ & \mathcal{M}' \text{ witnesses space } n^k \text{ for some } k \,\} \end{aligned}$$

Machine in constructible space s:

$\mathcal{M}$ is a Turing machine;

$\mathcal{M}'$ is a witness of s;

$\mathcal{M}'; \mathcal{M}$ is a new machine $\widehat{\mathcal{M}}$ marked with s.

Just modify the finite control of $\mathcal{M}$ to incorporate the finite control of $\mathcal{M}'$; a mark is put at cell $s(|x|)$ and all the other cells are cleaned.

Example of a complexity class:

$$\begin{aligned} PSPACE \quad = \quad & \{\, A \subseteq \{0,1\}^\star : \\ & A \text{ is decidable by } \mathcal{M}'; \mathcal{M}, \\ & \text{where } \mathcal{M} \text{ is a Turing machine and} \\ & \mathcal{M}' \text{ witnesses space } n^k \text{ for some } k \,\} \end{aligned}$$

Machine in constructible space s:

$\mathcal{M}$ is a Turing machine;

$\mathcal{M}'$ is a witness of s;

$\mathcal{M}'; \mathcal{M}$ is a new machine $\widehat{\mathcal{M}}$ marked with s.

Just modify the finite control of $\mathcal{M}$ to incorporate the finite control of $\mathcal{M}'$; a mark is put at cell $s(|x|)$ and all the other cells are cleaned.

Example of a complexity class:

$$\begin{aligned} PSPACE \quad = \quad & \{\ A \subseteq \{0,1\}^\star : \\ & A \text{ is decidable by } \mathcal{M}'; \mathcal{M}, \\ & \text{where } \mathcal{M} \text{ is a Turing machine and} \\ & \mathcal{M}' \text{ witnesses space } n^k \text{ for some } k\ \} \end{aligned}$$

Machine clocked in constructible time t:

$\mathcal{M}$ is a Turing machine;
$\mathcal{M}'$ is a witness of t;
$\mathcal{M}||\mathcal{M}'$ is a new machine $\widehat{\mathcal{M}}$ clocked in time t;
Just add the tapes of $\mathcal{M}'$ to $\mathcal{M}$ and modify the finite control of $\mathcal{M}$ to incorporate the clock $\mathcal{M}'$.

Example of a complexity class:

$$P = \{ A \subseteq \{0,1\}^* : A \text{ is decidable by } \mathcal{M}||\mathcal{M}', \text{ where } \mathcal{M} \text{ is a Turing machine and } \mathcal{M}' \text{ witnesses time } n^k \text{ for some } k \}$$

Machine clocked in constructible time t:

$\mathcal{M}$ is a Turing machine;

$\mathcal{M}'$ is a witness of t;

$\mathcal{M}||\mathcal{M}'$ is a new machine $\widehat{\mathcal{M}}$ clocked in time t;

Just add the tapes of $\mathcal{M}'$ to $\mathcal{M}$ and modify the finite control of $\mathcal{M}$ to incorporate the clock $\mathcal{M}'$.

Example of a complexity class:

$$P = \{ A \subseteq \{0,1\}^* : A \text{ is decidable by } \mathcal{M}||\mathcal{M}', \text{ where } \mathcal{M} \text{ is a Turing machine and } \mathcal{M}' \text{ witnesses time } n^k \text{ for some } k \}$$

Machine clocked in constructible time t:

$\mathcal{M}$ is a Turing machine;

$\mathcal{M}'$ is a witness of t;

$\mathcal{M}||\mathcal{M}'$ is a new machine $\widehat{\mathcal{M}}$ clocked in time t;

Just add the tapes of $\mathcal{M}'$ to $\mathcal{M}$ and modify the finite control of $\mathcal{M}$ to incorporate the clock $\mathcal{M}'$.

Example of a complexity class:

$$P = \{ A \subseteq \{0,1\}^* : A \text{ is decidable by } \mathcal{M}||\mathcal{M}', \text{ where } \mathcal{M} \text{ is a Turing machine and } \mathcal{M}' \text{ witnesses time } n^k \text{ for some } k \}$$

Machine clocked in constructible time t:

$\mathcal{M}$ is a Turing machine;

$\mathcal{M}'$ is a witness of t;

$\mathcal{M}||\mathcal{M}'$ is a new machine $\widehat{\mathcal{M}}$ clocked in time t;

Just add the tapes of $\mathcal{M}'$ to $\mathcal{M}$ and modify the finite control of $\mathcal{M}$ to incorporate the clock $\mathcal{M}'$.

Example of a complexity class:

$$P = \{ A \subseteq \{0,1\}^* : A \text{ is decidable by } \mathcal{M}||\mathcal{M}', \text{ where } \mathcal{M} \text{ is a Turing machine and } \mathcal{M}' \text{ witnesses time } n^k \text{ for some } k \}$$

Machine clocked in constructible time t:

$\mathcal{M}$ is a Turing machine;
$\mathcal{M}'$ is a witness of t;
$\mathcal{M}||\mathcal{M}'$ is a new machine $\widehat{\mathcal{M}}$ clocked in time t;
Just add the tapes of $\mathcal{M}'$ to $\mathcal{M}$ and modify the finite control of $\mathcal{M}$ to incorporate the clock $\mathcal{M}'$.

Example of a complexity class:

$$P = \{ A \subseteq \{0,1\}^* : A \text{ is decidable by } \mathcal{M}||\mathcal{M}', \text{ where } \mathcal{M} \text{ is a Turing machine and } \mathcal{M}' \text{ witnesses time } n^k \text{ for some } k \}$$

Machine clocked in constructible time t:

$\mathcal{M}$ is a Turing machine;
$\mathcal{M}'$ is a witness of t;
$\mathcal{M}||\mathcal{M}'$ is a new machine $\widehat{\mathcal{M}}$ clocked in time t;
Just add the tapes of $\mathcal{M}'$ to $\mathcal{M}$ and modify the finite control of $\mathcal{M}$ to incorporate the clock $\mathcal{M}'$.

Example of a complexity class:

$$
\begin{aligned}
P = \quad & \{ \\
& A \subseteq \{0,1\}^\star : \\
& A \text{ is decidable by } \mathcal{M}||\mathcal{M}', \\
& \text{where } \mathcal{M} \text{ is a Turing machine and} \\
& \mathcal{M}' \text{ witnesses time } n^k \text{ for some } k \\
& \}
\end{aligned}
$$

Halting Problem in bounded space

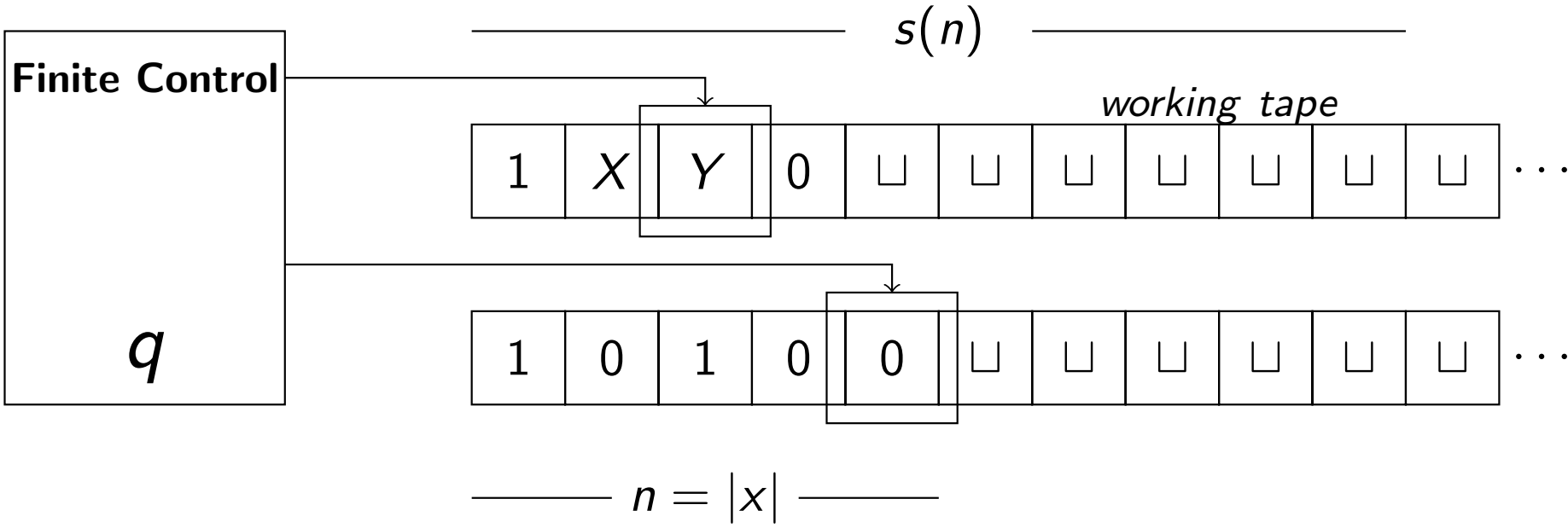
Theorem

The halting problem of Turing machines bounded in space is decidable.

Proof

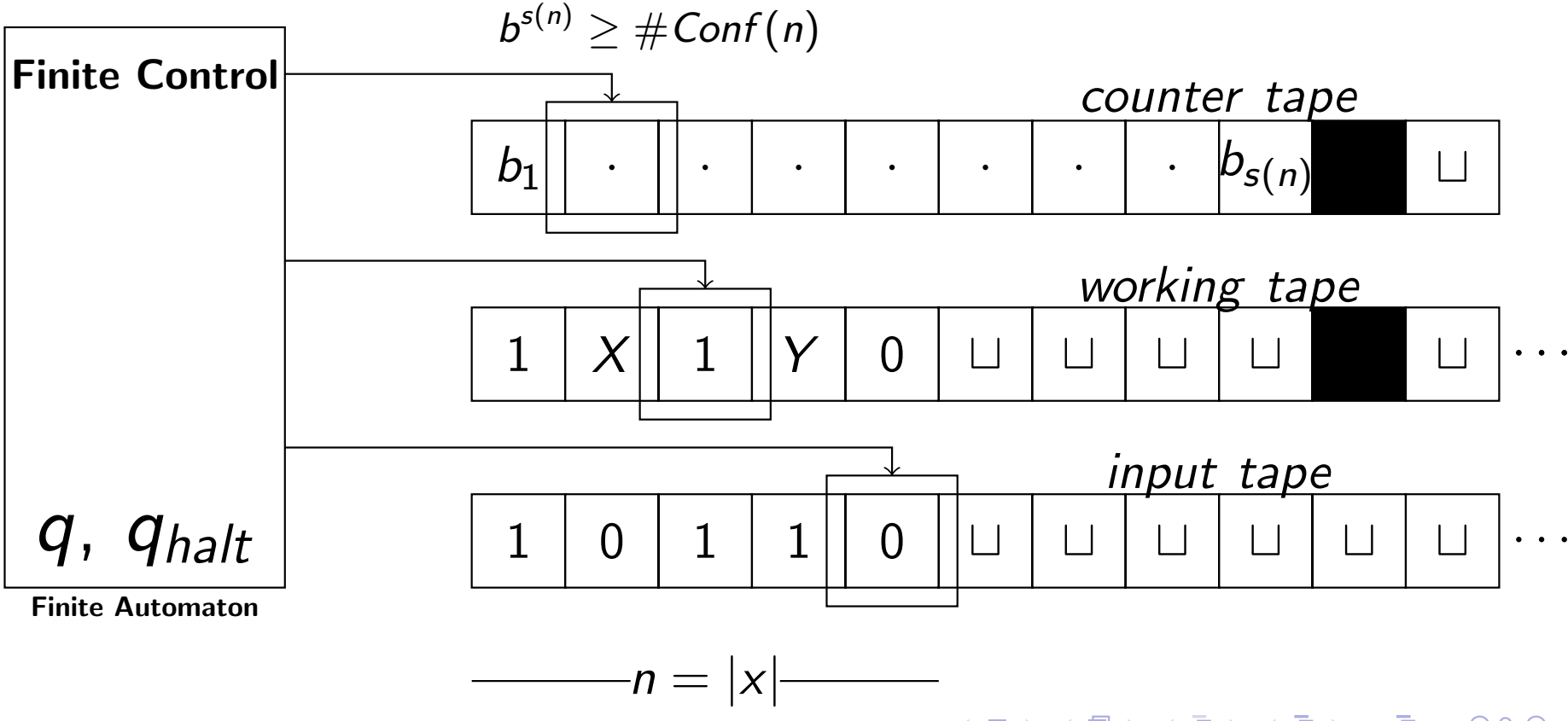
$$\begin{aligned} \#Configurations(n) &= |Q| \times 3^{s(n)} \times s(n) \times n \\ &\in O(2^{O(s(n)} \times n) \\ &= 2^{O(s(n))} \\ &= b^{s(n)} \end{aligned}$$

Halting Problem in bounded space



Halting Problem in bounded space

We add a new tape to count in base b.



Conclusions of this section

The halting problem of Turing machines is decidable in bounded space;

The halting problem becomes interesting only when all the infinite tape is potentially used;

These are the important facts to take into consideration when we try to embed an infinite tape Turing machine into the physical space.

Conclusions of this section

The halting problem of Turing machines is decidable in bounded space;

The halting problem becomes interesting only when all the infinite tape is potentially used;

These are the important facts to take into consideration when we try to embed an infinite tape Turing machine into the physical space.

Conclusions of this section

The halting problem of Turing machines is decidable in bounded space;

The halting problem becomes interesting only when all the infinite tape is potentially used;

These are the important facts to take into consideration when we try to embed an infinite tape Turing machine into the physical space.

Undecidability in Analysis

Undecidability in Analysis

Let $\mathcal{E}$ be a set of expressions denoting real, single valued, partially defined functions of one variable and let Φ be the set of functions denoted by expressions in $\mathcal{E}$.

Notation

$A(x)$ is the function denoted by A.

$A(x) \equiv B(x)$ means that $A(x)$ and $B(x)$ are defined at the same points and are equal whenever they are defined.

Undecidability in Analysis

Let $\mathcal{E}$ be a set of expressions denoting real, single valued, partially defined functions of one variable and let Φ be the set of functions denoted by expressions in $\mathcal{E}$.

Notation

$A(x)$ is the function denoted by A.

$A(x) \equiv B(x)$ means that $A(x)$ and $B(x)$ are defined at the same points and are equal whenever they are defined.

Undecidability in Analysis

Let $\mathcal{E}$ be a set of expressions denoting real, single valued, partially defined functions of one variable and let Φ be the set of functions denoted by expressions in $\mathcal{E}$.

Notation

$A(x)$ is the function denoted by A.

$A(x) \equiv B(x)$ means that $A(x)$ and $B(x)$ are defined at the same points and are equal whenever they are defined.

Assumptions

Φ contains the identity function, the rational numbers, and is closed under addition, subtraction, multiplication, and composition;

If $A, B \in \mathcal{E}$, then there is an effective procedure for finding expressions in $\mathcal{E}$ to denote $A(x) \pm B(x)$, $A(x) \times B(x)$, $A(B(x))$.

Assumptions

Φ *contains the identity function, the rational numbers, and is closed under addition, subtraction, multiplication, and composition;*

If $A, B \in \mathcal{E}$*, then there is an effective procedure for finding expressions in* $\mathcal{E}$ *to denote* $A(x) \pm B(x)$*,* $A(x) \times B(x)$*,* $A(B(x))$*.*

Assumptions

Φ *contains the identity function, the rational numbers, and is closed under addition, subtraction, multiplication, and composition;*

If $A, B \in \mathcal{E}$*, then there is an effective procedure for finding expressions in* $\mathcal{E}$ *to denote* $A(x) \pm B(x)$, $A(x) \times B(x)$, $A(B(x))$.

Problems to be discussed

The identity problem for $(\mathcal{E}, \Phi)$ is the problem of deciding, given $A \in \mathcal{E}$, whether $A(x) \equiv 0$.

The integration problem for $(\mathcal{E}, \Phi)$ is the problem of deciding, given $A \in \mathcal{E}$, whether there is function $f \in \Phi$ such that $f'(x) \equiv A(x)$.

Problems to be discussed

The identity problem for $(\mathcal{E}, \Phi)$ is the problem of deciding, given $A \in \mathcal{E}$, whether $A(x) \equiv 0$.

The integration problem for $(\mathcal{E}, \Phi)$ is the problem of deciding, given $A \in \mathcal{E}$, whether there is function $f \in \Phi$ such that $f'(x) \equiv A(x)$.

Problems to be discussed

The identity problem for $(\mathcal{E}, \Phi)$ *is the problem of deciding, given* $A \in \mathcal{E}$*, whether* $A(x) \equiv 0$*.*

The integration problem for $(\mathcal{E}, \Phi)$ *is the problem of deciding, given* $A \in \mathcal{E}$*, whether there is function* $f \in \Phi$ *such that* $f'(x) \equiv A(x)$*.*

Conditions on Φ

1. Φ *contains* π *and the real-valued function* $\sin(x)$*;*
2. Φ *contains* μ *such that* $\mu(x) = |x|$ *for* $x \neq 0$*;*
3. Φ *contains* β*, a totally defined function such that no* $f \in \Phi$ *and no interval* $\mathcal{I}$ *are such that* $f' \equiv \beta$ *in* $\mathcal{I}$*.*

Conditions on Φ

1. Φ *contains* π *and the real-valued function* $\sin(x)$*;*
2. Φ *contains* μ *such that* $\mu(x) = |x|$ *for* $x \neq 0$*;*
3. Φ *contains* β*, a totally defined function such that no* $f \in \Phi$ *and no interval* $\mathcal{I}$ *are such that* $f' \equiv \beta$ *in* $\mathcal{I}$*.*

Conditions on Φ

1. Φ *contains* π *and the real-valued function* $\sin(x)$*;*
2. Φ *contains* μ *such that* $\mu(x) = |x|$ *for* $x \neq 0$*;*
3. Φ *contains* β*, a totally defined function such that no* $f \in \Phi$ *and no interval* $\mathcal{I}$ *are such that* $f' \equiv \beta$ *in* $\mathcal{I}$*.*

Conditions on Φ

1. Φ *contains* π *and the real-valued function* $\sin(x)$*;*
2. Φ *contains* μ *such that* $\mu(x) = |x|$ *for* $x \neq 0$*;*
3. Φ *contains* β*, a totally defined function such that no* $f \in \Phi$ *and no interval* $\mathcal{I}$ *are such that* $f' \equiv \beta$ *in* $\mathcal{I}$*.*

Theorem

1. *If* Φ *satisfies condition* 1, *then the problem* given an expression $A \in \mathcal{E}$, decide if there is a real number x such that $A(x) < 0$ *is undecidable.*

2. *If* Φ *satisfies conditions* 1 *and* 2, *then the identity problem for* $(\mathcal{E}, \Phi)$ *is undecidable.*

3. *If* Φ *satisfies conditions* 1, 2 *and* 3, *then the integration problem for* $(\mathcal{E}, \Phi)$ *is undecidable.*

Theorem

1. *If* Φ *satisfies condition* 1, *then the problem* given an expression $A \in \mathcal{E}$, decide if there is a real number x such that $A(x) < 0$ *is undecidable.*
2. *If* Φ *satisfies conditions* 1 *and* 2, *then the identity problem for* $(\mathcal{E}, \Phi)$ *is undecidable.*
3. *If* Φ *satisfies conditions* 1, 2 *and* 3, *then the integration problem for* $(\mathcal{E}, \Phi)$ *is undecidable.*

Theorem

1. *If* Φ *satisfies condition* 1, *then the problem* given an expression $A \in \mathcal{E}$, decide if there is a real number x such that $A(x) < 0$ *is undecidable.*

2. *If* Φ *satisfies conditions* 1 *and* 2, *then the identity problem for* $(\mathcal{E}, \Phi)$ *is undecidable.*

3. *If* Φ *satisfies conditions* 1, 2 *and* 3, *then the integration problem for* $(\mathcal{E}, \Phi)$ *is undecidable.*

Theorem

1. *If* Φ *satisfies condition* 1, *then the problem* given an expression $A \in \mathcal{E}$, decide if there is a real number x such that $A(x) < 0$ *is undecidable.*

2. *If* Φ *satisfies conditions* 1 *and* 2, *then the identity problem for* $(\mathcal{E}, \Phi)$ *is undecidable.*

3. *If* Φ *satisfies conditions* 1, 2 *and* 3, *then the integration problem for* $(\mathcal{E}, \Phi)$ *is undecidable.*

Existence

- $\mathcal{E}$ is the smallest class of expressions obtained by iteration of addition, subtraction, multiplication, and composition, starting with x, e^x, $\sin(x)$ and $|x|$, and expressions for the rational numbers;
- Φ is the class of functions of a real variable usually denoted by the expressions above; take $\beta(x) = e^{x^2}$ and $\mu(x) = |x|$.

Existence

- $\mathcal{E}$ is the smallest class of expressions obtained by iteration of addition, subtraction, multiplication, and composition, starting with x, e^x, $\sin(x)$ and $|x|$, and expressions for the rational numbers;
- Φ is the class of functions of a real variable usually denoted by the expressions above; take $\beta(x) = e^{x^2}$ and $\mu(x) = |x|$.

Existence

- $\mathcal{E}$ is the smallest class of expressions obtained by iteration of addition, subtraction, multiplication, and composition, starting with x, e^x, $\sin(x)$ and $|x|$, and expressions for the rational numbers;
- Φ is the class of functions of a real variable usually denoted by the expressions above; take $\beta(x) = e^{x^2}$ and $\mu(x) = |x|$.

Algebra of subelementary expressions

Let $\mathcal{A}$ be the smallest class containing expressions for the rational numbers and π, and, for all i, x_i, and $\sin(x_i)$, and closed under addition, subtraction, multiplication, and composition.

Theorem $\mathcal{A}$ is closed under differentiation.

Algebra of subelementary expressions

Let $\mathcal{A}$ be the smallest class containing expressions for the rational numbers and π, and, for all i, x_i, and $\sin(x_i)$, and closed under addition, subtraction, multiplication, and composition.

Theorem $\mathcal{A}$ is closed under differentiation.

Algebra of subelementary expressions

Let $\mathcal{A}$ be the smallest class containing expressions for the rational numbers and π, and, for all i, x_i, and $\sin(x_i)$, and closed under addition, subtraction, multiplication, and composition.

Theorem $\mathcal{A}$ is closed under differentiation.

On the making

Theorem

For any subelementary function $f(x_1, \ldots, x_n)$, it is possible to find a subelementary function $g^{[f]}(x_1, \ldots, x_n)$ so that:

$$\begin{aligned} \forall x_1, \ldots, x_n \in \mathbb{R} \quad & g^{[f]}(x_1, \ldots, x_n) > 1 \,, \\ \forall x_1, \ldots, x_n, \delta_1, \ldots, \delta_n \in \mathbb{R} \quad & |\delta_1| \leq 1 \wedge \cdots \wedge |\delta_n| \leq 1 \implies \\ & g^{[f]}(x_1, \ldots, x_n) > |f(x_1 + \delta_1, \ldots, x_n + \delta_n)| \,. \end{aligned}$$

On the making

Theorem

For any subelementary function $f(x_1, \ldots, x_n)$, it is possible to find a subelementary function $g^{[f]}(x_1, \ldots, x_n)$ so that:

$$\begin{aligned} \forall x_1, \ldots, x_n \in \mathbb{R} \quad & g^{[f]}(x_1, \ldots, x_n) > 1 \, , \\ \forall x_1, \ldots, x_n, \delta_1, \ldots, \delta_n \in \mathbb{R} \quad & |\delta_1| \leq 1 \wedge \cdots \wedge |\delta_n| \leq 1 \implies \\ & g^{[f]}(x_1, \ldots, x_n) > |f(x_1 + \delta_1, \ldots, x_n + \delta_n)| \, . \end{aligned}$$

On the making

Theorem

For any subelementary function $f(x_1, \ldots, x_n)$, it is possible to find a subelementary function $g^{[f]}(x_1, \ldots, x_n)$ so that:

$$\begin{aligned} \forall x_1, \ldots, x_n \in \mathbb{R} \qquad & g^{[f]}(x_1, \ldots, x_n) \; > \; 1 \; , \\ \forall x_1, \ldots, x_n, \delta_1, \ldots, \delta_n \in \mathbb{R} \qquad & |\delta_1| \leq 1 \wedge \cdots \wedge |\delta_n| \leq 1 \implies \\ & g^{[f]}(x_1, \ldots, x_n) \; > \; |f(x_1 + \delta_1, \ldots, x_n + \delta_n)| \; . \end{aligned}$$

On the making

$$
\begin{aligned}
h(x) &= x\sin(x) \\
g(x) &= x\sin(x^3) \\
x_1 &= h(x) \\
x_2 &= h \circ g(x) \\
x_3 &= h \circ g \circ g(x) \\
&\dots \\
x_{n-1} &= h \circ \overbrace{g \circ \dots \circ g}^{n-2}(x) \\
x_n &= \overbrace{g \circ \dots \circ g}^{n}(x)
\end{aligned}
$$

On the making

$$
\begin{aligned}
f^{[p]}(m, x_1, \ldots, x_n) &= (n+1)^4 \{ p(m, x_1, \ldots, x_n)^2 \\
&\quad + \sum_{i=1}^{n} \sin^2(\pi x_i) \; [\; g^{[\partial_{x_i} p(m, x_1, \ldots, x_n)^2]}(m, x_1, \ldots, x_n) \;]^4 \} \\
F^{[p]}(m, x_1, \ldots, x_n) &= f(m, x_1^2, \ldots, x_n^2) \\
G^{[p]}(m, x) &= F^{[p]}(m, x_1(x), \ldots, x_n(x)) \; .
\end{aligned}
$$

Theorem

$$\begin{aligned} \exists x_1, \ldots, x_n \in \mathbb{N}\ p(m, x_1, \ldots, x_n) &= 0 \\ &\textit{iff} \\ \exists x_1, \ldots, x_n \in \mathbb{R}\ F^{[p]}(m, x_1, \ldots, x_n) &= 0 \\ &\textit{iff} \\ \exists x_1, \ldots, x_n \in \mathbb{R}\ F^{[p]}(m, x_1, \ldots, x_n) &\leq 1 \end{aligned}$$

Theorem

$$\begin{aligned} \exists x_1, \ldots, x_n \in \mathbb{N}\ p(m, x_1, \ldots, x_n) &= 0 \\ &\textit{iff} \\ \exists x \in \mathbb{R}\ G^{[p]}(m, x) &\leq 1 \\ &\textit{iff} \\ \forall z > 0\ \exists x \in \mathbb{R}\ G^{[p]}(m, x) &\leq z \end{aligned}$$

Theorem (Richardson, 1968)

There is a subelementary function of two variables, $G(m, x)$, such that, as m varies over $\mathbb{N}$, there is no algorithm for deciding whether is a real number x such that $G(m, x) \leq 1$.

Theorem

$$\begin{aligned} \exists x_1, \ldots, x_n \in \mathbb{N}\ p(m, x_1, \ldots, x_n) &= 0 \\ &\textit{iff} \\ \exists x \in \mathbb{R}\ G^{[p]}(m, x) &\leq 1 \\ &\textit{iff} \\ \forall z > 0\ \exists x \in \mathbb{R}\ G^{[p]}(m, x) &\leq z \end{aligned}$$

Theorem (Richardson, 1968)

There is a subelementary function of two variables, $G(m, x)$, such that, as m varies over $\mathbb{N}$, there is no algorithm for deciding whether is a real number x such that $G(m, x) \leq 1$.

Theorem (Richardson, 1968)

If Φ contains the identity function, the rational numbers, π, the real-valued functions of expressions $|x|$ and $\sin(x)$, and is closed under addition, subtraction, multiplication, and composition, then the identity problem for $(\mathcal{E}, \Phi)$ is undecidable.

Proof Take $B(m,x) = |G(m,x)-1| - (G(m,x)-1)$. We have that $\exists x\ G(m,x) < 1$ if and only if $B(m,x) \not\equiv 0$.

Theorem (Richardson, 1968)

If Φ contains the identity function, the rational numbers, π, the real-valued functions of expressions $|x|$ and $\sin(x)$, and is closed under addition, subtraction, multiplication, and composition, then the identity problem for $(\mathcal{E}, \Phi)$ is undecidable.

Proof Take $B(m,x) = |G(m,x)-1| - (G(m,x)-1)$. We have that $\exists x\ G(m,x) < 1$ if and only if $B(m,x) \not\equiv 0$.

Theorem (Richardson, 1968)

If Φ contains the identity function, the rational numbers, π, the real-valued functions of expressions $|x|$ and $\sin(x)$, and is closed under addition, subtraction, multiplication, and composition, then the identity problem for $(\mathcal{E}, \Phi)$ is undecidable.

Proof Take $B(m,x) = |G(m,x) - 1| - (G(m,x) - 1)$. We have that $\exists x\ G(m,x) < 1$ if and only if $B(m,x) \not\equiv 0$.

Theorem (Richardson, 1968)

Theorem *If Φ contains the identity function, the rational numbers, π, the real-valued functions of expressions $|x|$, e^x and $\sin(x)$, and it is closed under addition, subtraction, multiplication, and composition, then the integration problem for $(\mathcal{E}, \Phi)$ is undecidable.*

Proof If such integration problem were solvable, we would be able to decide, for each $m \in \mathbb{N}$, whether there were a function $f \in \Phi$ so that

$$f'(x) \;=\; e^{x^2}\,(1 - (1 \dot{-} (2 \dot{-} 2G(m,x)))\,.$$

Theorem (Richardson, 1968)

Theorem *If Φ contains the identity function, the rational numbers, π, the real-valued functions of expressions $|x|$, e^x and $\sin(x)$, and it is closed under addition, subtraction, multiplication, and composition, then the integration problem for $(\mathcal{E}, \Phi)$ is undecidable.*

Proof If such integration problem were solvable, we would be able to decide, for each $m \in \mathbb{N}$, whether there were a function $f \in \Phi$ so that

$$f'(x) = e^{x^2}\,(1 - (1 \dot{-} (2 \dot{-} 2G(m,x)))\,.$$

Theorem (Richardson, 1968)

Theorem *If Φ contains the identity function, the rational numbers, π, the real-valued functions of expressions $|x|$, e^x and $\sin(x)$, and it is closed under addition, subtraction, multiplication, and composition, then the integration problem for $(\mathcal{E}, \Phi)$ is undecidable.*

Proof If such integration problem were solvable, we would be able to decide, for each $m \in \mathbb{N}$, whether there were a function $f \in \Phi$ so that

$$f'(x) \;=\; e^{x^2}\,(1-(1\dot{-}(2\dot{-}2G(m,x)))\;.$$

Undecidability in Physics

Undecidability in Physics

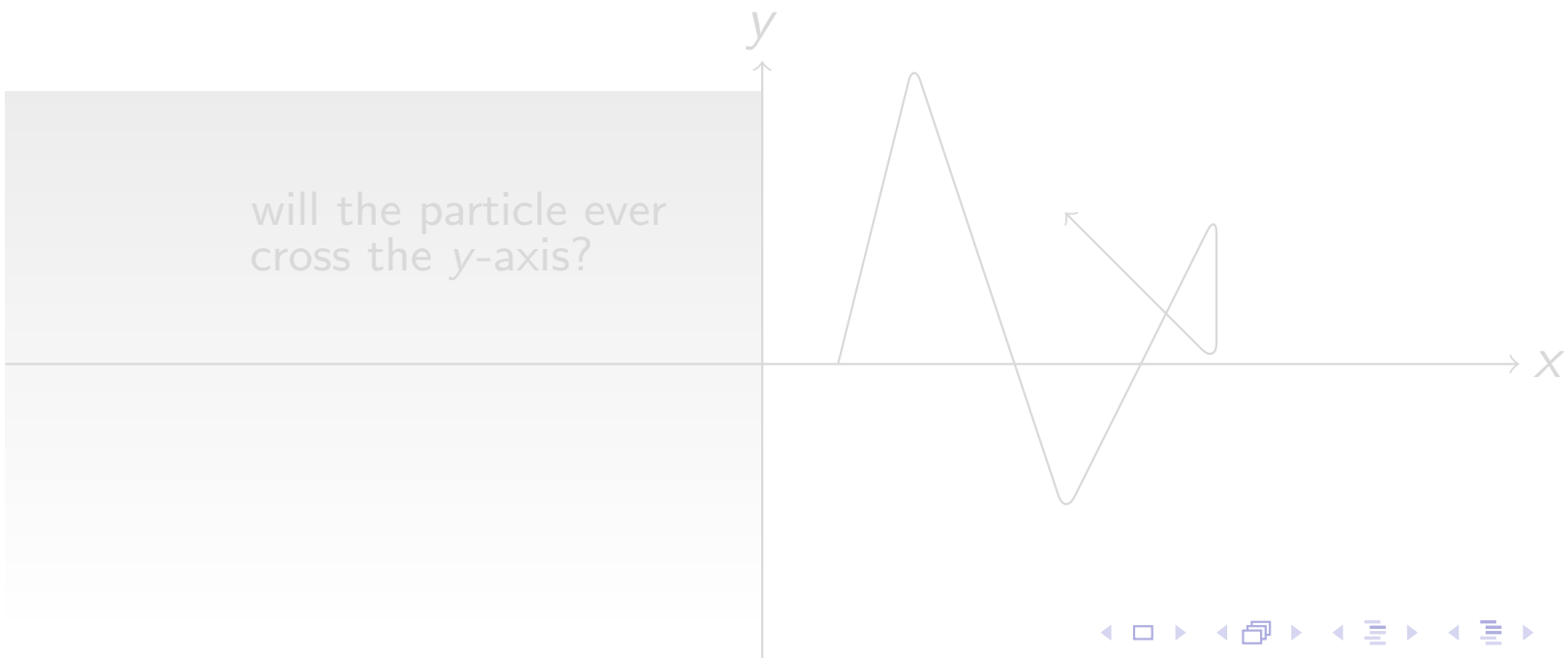
Are there general methods to test for the integrability of a given Hamiltonian? The answer, for the moment, is no. We can turn the question around, however, and ask if methods can be found to construct potentials that give rise to integrable Hamiltonians. The answer is that a method exists, at least for restricted class of problems, but the method becomes rapidly very tedious as the forms allowed for the integrals of the motion are expanded. *(A. J. Lichtenberg and M. A. Liberman,* Regular and Stochastic Motion.*)*

Undecidability in Physics

A recurrent theme in the preceding chapters has been the distinction between integrable and non-integrable systems. The later can ehxibit chaotic behaviour, whereras the former are distinguished by the existence of a full complement [...]. Despite all the advances in non-linear dynamics, a fundamental question still remains; mainly: given a system of equations, how can one tell a priori whether or not they are integrable? *(M. Tabor,* Chaos and Integrability in Nonlinear Dynamics*.)*

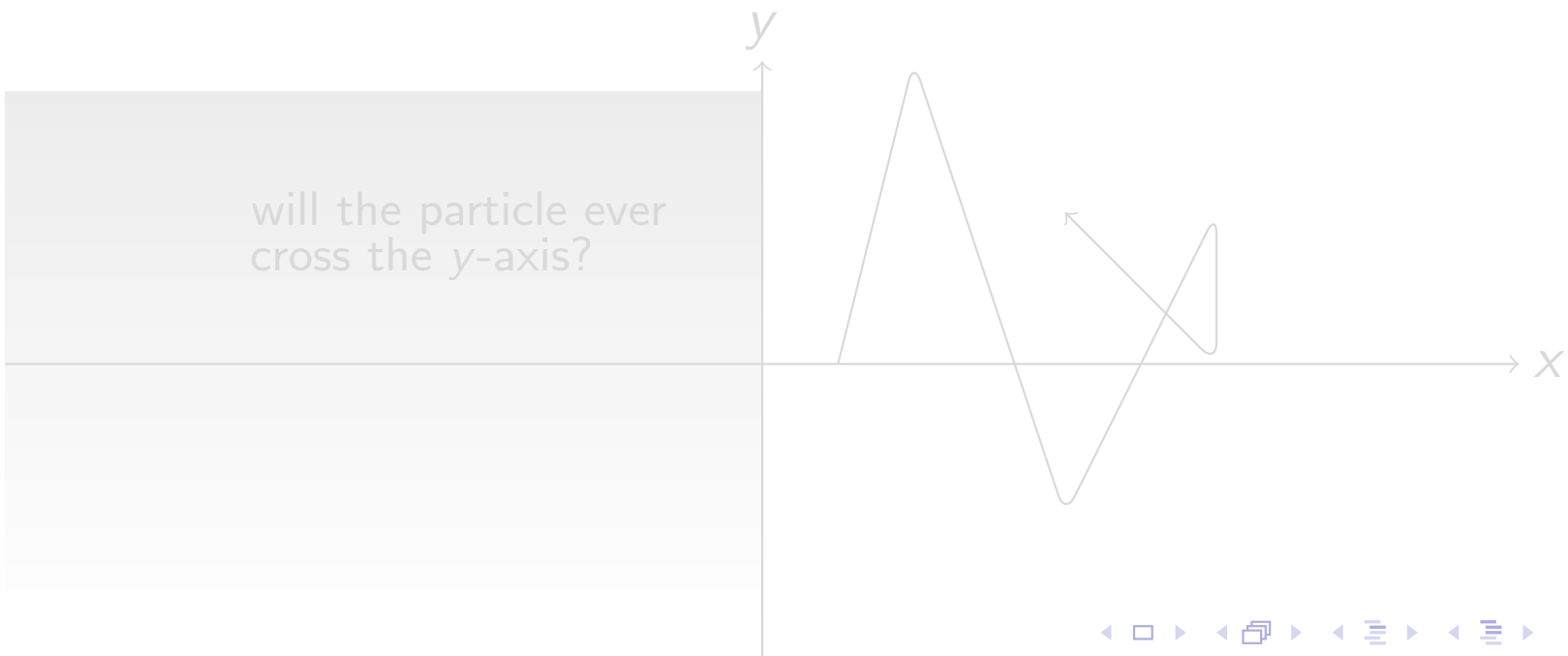
Theorem

Theorem *There is no general algorithmic procedure to determine whether an arbitrary motion in the $\langle x, y\rangle$-plane (analytical map $m : \mathbb{R} \to \mathbb{R}^2$), $m(t) = \langle x(t), y(t)\rangle$ will cross the y-axis.*



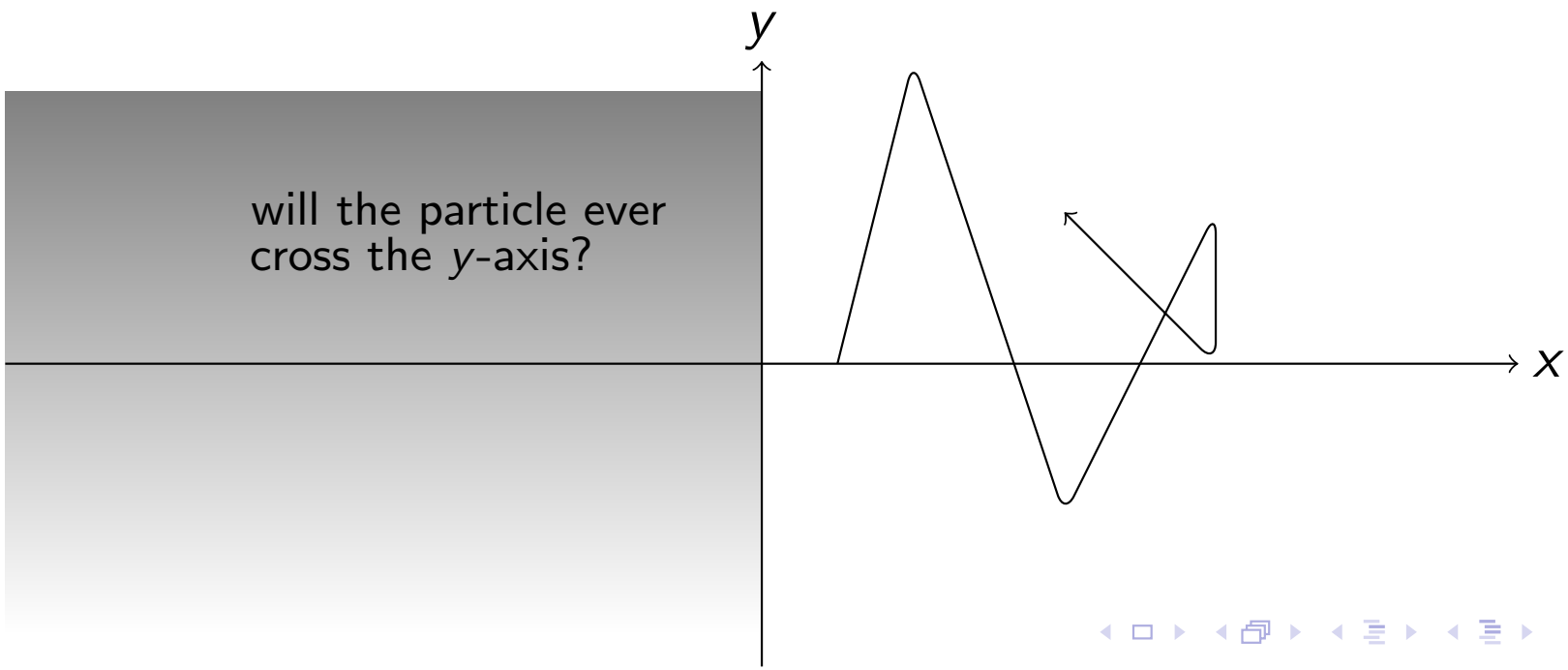
Theorem

Theorem *There is no general algorithmic procedure to determine whether an arbitrary motion in the $\langle x, y\rangle$-plane (analytical map $m : \mathbb{R} \to \mathbb{R}^2$), $m(t) = \langle x(t), y(t)\rangle$ will cross the y-axis.*



Theorem

Theorem *There is no general algorithmic procedure to determine whether an arbitrary motion in the $\langle x, y\rangle$-plane (analytical map $m : \mathbb{R} \to \mathbb{R}^2$), $m(t) = \langle x(t), y(t)\rangle$ will cross the y-axis.*



Motion in the plane

Proof Take $x_m(t) = G(m,t) - 1$. There is no general decision procedure to check whether one has, given an arbitrary $m \in \mathbb{N}$, $x_m(t) < 0$ for some t. Take $m(t) = \langle x_m(t), \frac{1}{2}gt^2 \rangle$.

Motion in the plane

Proof Take $x_m(t) = G(m,t) - 1$. There is no general decision procedure to check whether one has, given an arbitrary $m \in \mathbb{N}$, $x_m(t) < 0$ for some t. Take $m(t) = \langle x_m(t), \frac{1}{2}gt^2 \rangle$.

Hamiltonians [da Costa and Dória 1990-2]

$$\theta(m) = \begin{cases} 0 & \text{if } G(m,x) > 1, \text{ for all } x \\ 1 & \text{if } G(m,x) \leq 1/2, \text{ for some } x \end{cases}$$

$$B(m,x) = |G(m,x) - 1| - (G(m,x) - 1)$$

$$D(m) = \int_{-\infty}^{+\infty} \frac{B(m,x)e^{-x^2}}{1 + B(m,x)}\, dx$$

$$\theta(m) = 1 - \delta\left(\frac{D(m)}{1 + D(m)}\right) .$$

Theorem (da Costa and Dória 1990-2)

There is no general decision procedure to check whether, for a given $m \in \mathbb{N}$,

- *There is a real number x such that $G(m, x) \leq 1$;*
- *We have $\theta(m) = 0$ or $\theta(m) = 1$.*

Theorem (da Costa and Dória 1990-2)

There is no general decision procedure to check whether, for a given $m \in \mathbb{N}$,

- *There is a real number x such that $G(m, x) \leq 1$;*
- *We have $\theta(m) = 0$ or $\theta(m) = 1$.*

Theorem (da Costa and Dória 1990-2)

There is no general decision procedure to check whether, for a given $m \in \mathbb{N}$,

- *There is a real number x such that $G(m, x) \leq 1$;*
- *We have $\theta(m) = 0$ or $\theta(m) = 1$.*

Theorem (da Costa and Dória 1990-2)

There is no general decision procedure to check whether, for a given $m \in \mathbb{N}$,

- *There is a real number* x *such that* $G(m, x) \leq 1$;
- *We have* $\theta(m) = 0$ *or* $\theta(m) = 1$.

Given an arbitrary Hamiltonian h, is there an algorithm that allows us to decide whether the associated Hamiltonian system X_h is integrable by quadratures?

Theorem (da Costa and Dória 1990-2)

There is no general algorithm to decide whether an arbitrary Hamiltonian system over U can be solved by quadratures.

Proof Consider the family of Hamiltonians given by

$$H_k(q,p) = \underbrace{1/2(p^2+q^2)}_{\text{harmonic oscilator}}\ \theta(k) + \underbrace{p}_{\text{free particle}}\ (1-\theta(k))\ .$$

Given an arbitrary Hamiltonian h, is there an algorithm that allows us to decide whether the associated Hamiltonian system X_h is integrable by quadratures?

Theorem (da Costa and Dória 1990-2)

There is no general algorithm to decide whether an arbitrary Hamiltonian system over U can be solved by quadratures.

Proof Consider the family of Hamiltonians given by

$$H_k(q,p) = \underbrace{1/2(p^2+q^2)}_{\text{harmonic oscilator}}\ \theta(k) + \underbrace{p}_{\text{free particle}}\ (1-\theta(k))\ .$$

Given an arbitrary Hamiltonian h, is there an algorithm that allows us to decide whether the associated Hamiltonian system X_h is integrable by quadratures?

Theorem (da Costa and Dória 1990-2)

There is no general algorithm to decide whether an arbitrary Hamiltonian system over U can be solved by quadratures.

Proof Consider the family of Hamiltonians given by

$$H_k(q,p) = \underbrace{1/2(p^2+q^2)}_{\text{harmonic oscilator}}\ \theta(k) + \underbrace{p}_{\text{free particle}}\ (1-\theta(k))\ .$$

Off Infinite in Finite Time

Off Infinite in Finite Time

Singularity A singularity is a time value $t = t^\star$ where analytic continuation of the solution fails. It requires a distance $r_{ij}(t)$ to become arbitrarily small as $t \to t^\star$. (E.g., a collision is a singularity. But are all singularities collisions? Problem raised in the turn XX/XXI by Painlevé and Zeipel. The way to the solution was provided by Sundman, Wintner, McGehee, Gerver, Saari, Xia.)

Off Infinite in Finite Time

Singularity A singularity is a time value $t = t^\star$ where analytic continuation of the solution fails. It requires a distance $r_{ij}(t)$ to become arbitrarilly small as $t \to t^\star$. (E.g., a collision is a singularity. But are all singularities collisions? Problem raised in the turn XX/XXI by Painlevé and Zeipel. The way to the solution was provided by Sundman, Wintner, McGehee, Gerver, Saari, Xia.)

Singularity

Let

- $r_{\min}(t) = \min_{i \neq j} r_{ij}(t)$.

We can have

- $\liminf_{t \to t^\star} r_{\min}(t) = 0$.
- $\limsup_{t \to t^\star} r_{\min}(t) > r > 0$.

Singularity

Let

- $r_{\min}(t) = \min_{i \neq j} r_{ij}(t)$.

We can have

- $\liminf_{t \to t^\star} r_{\min}(t) = 0$.
- $\limsup_{t \to t^\star} r_{\min}(t) > r > 0$.

Singularity

Let

- $r_{\min}(t) = \min_{i \neq j} r_{ij}(t)$.

We can have

- $\liminf_{t \to t^\star} r_{\min}(t) = 0$.
- $\limsup_{t \to t^\star} r_{\min}(t) > r > 0$.

Singularity

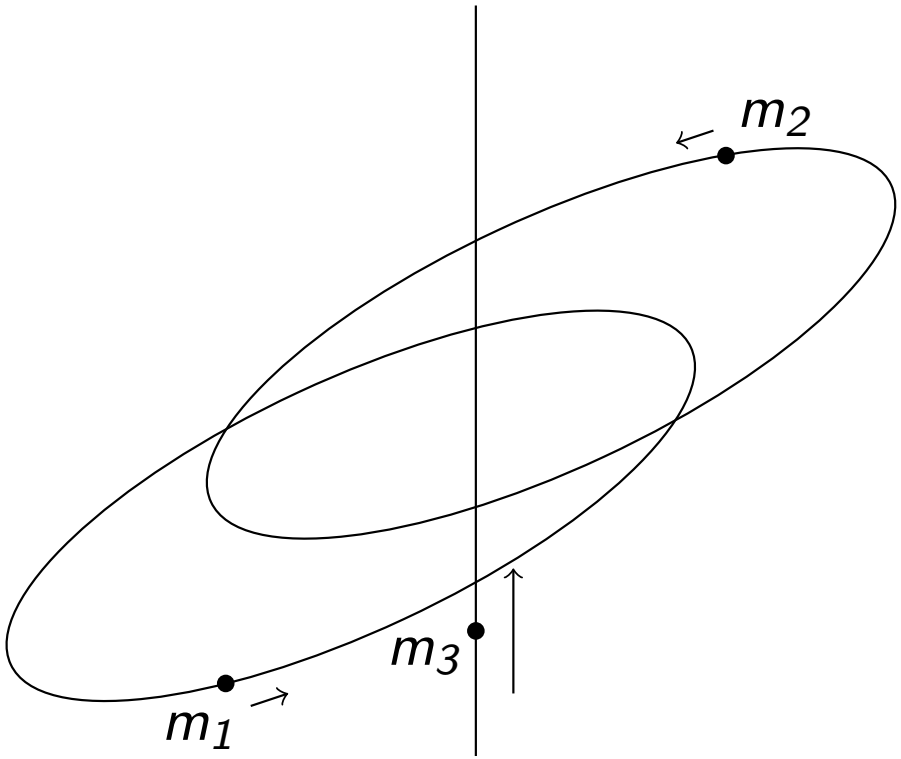
Let

- $r_{\min}(t) = \min_{i \neq j} r_{ij}(t)$.

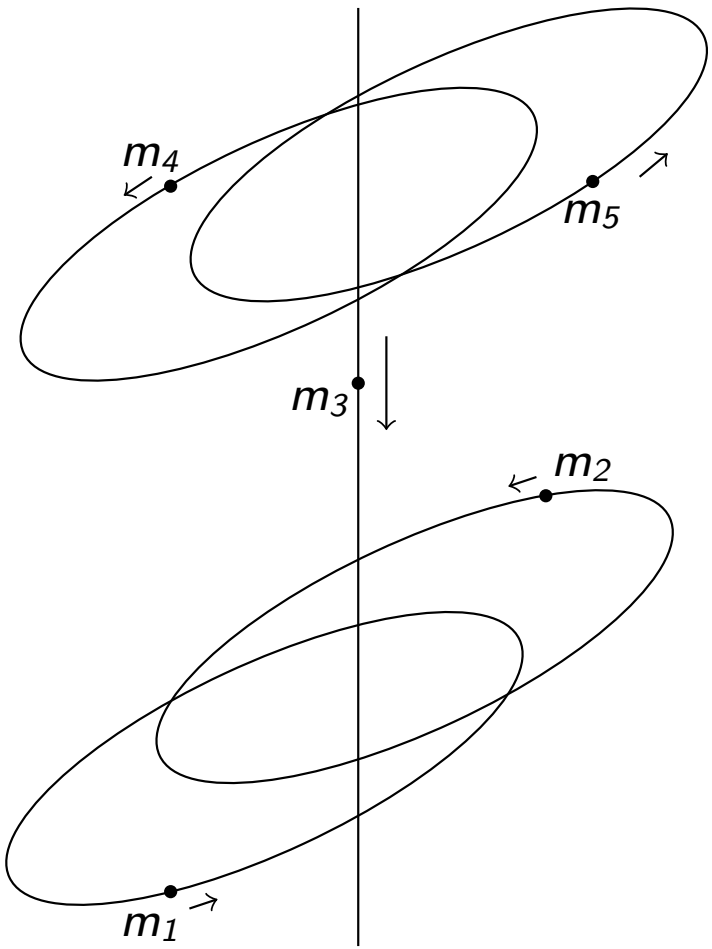
We can have

- $\liminf_{t \to t^\star} r_{\min}(t) = 0$.
- $\limsup_{t \to t^\star} r_{\min}(t) > r > 0$.

Singularity [Xia, 1992]

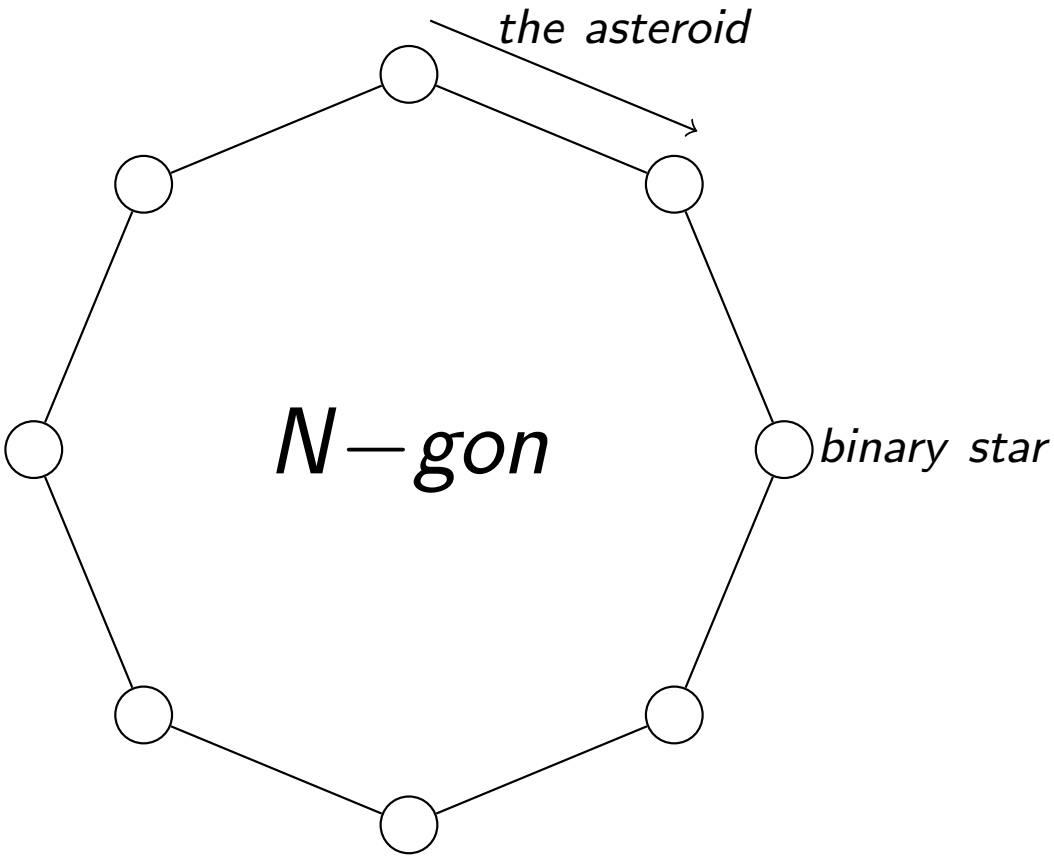


Singularity [Xia 1992]



Singularity [Gerver 1991]

Singularity [Gerver 1991]

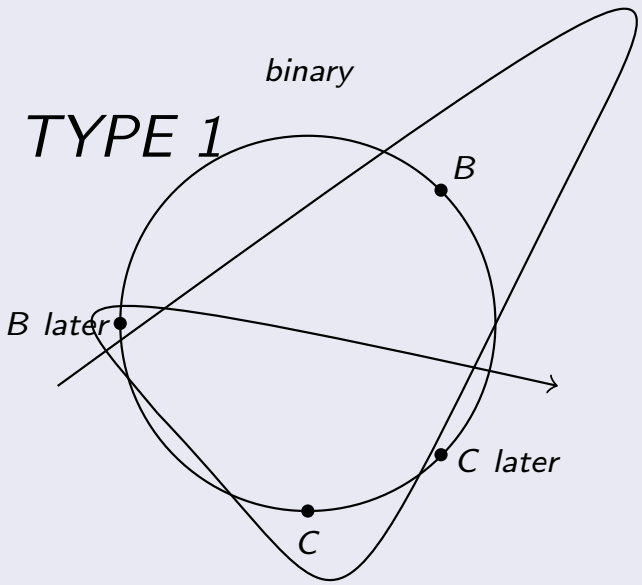


Singularity [Smith 2006]

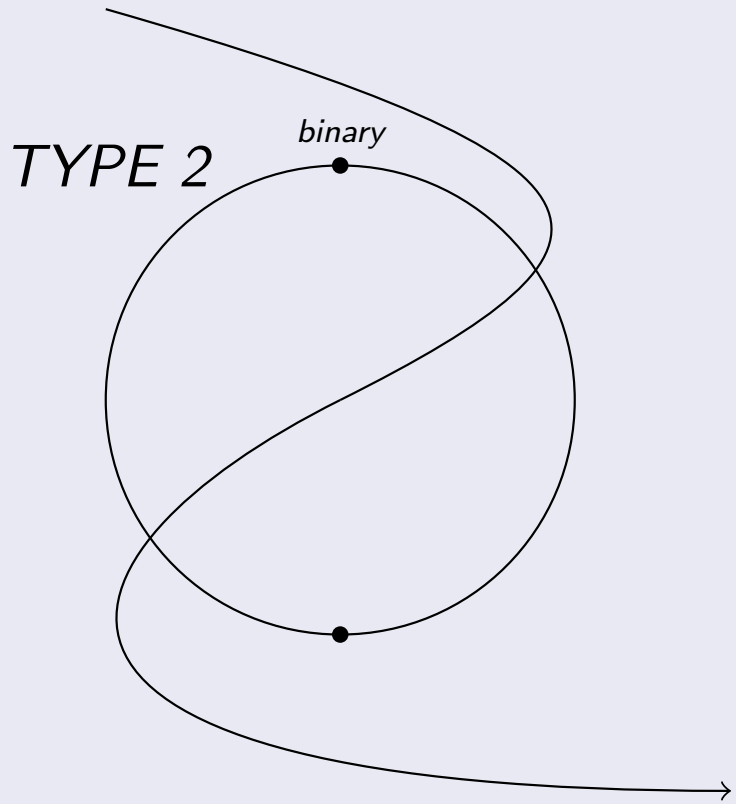
Theorem

N point masses in the plane, for some finite fixed value of N, whose initial positions, masses, and velocities lie inside a cube in $\mathbb{R}^{7n}$, can describe an uncountably infinite number of topological distinct trajectories in 1 second. In contrast, a Turing machine simulator can only output one of a finite number of possible outputs, in a finite timespan. The initial location and velocities of the bodies required to force a future trajectory of described topological type, are computable real numbers.

Topology

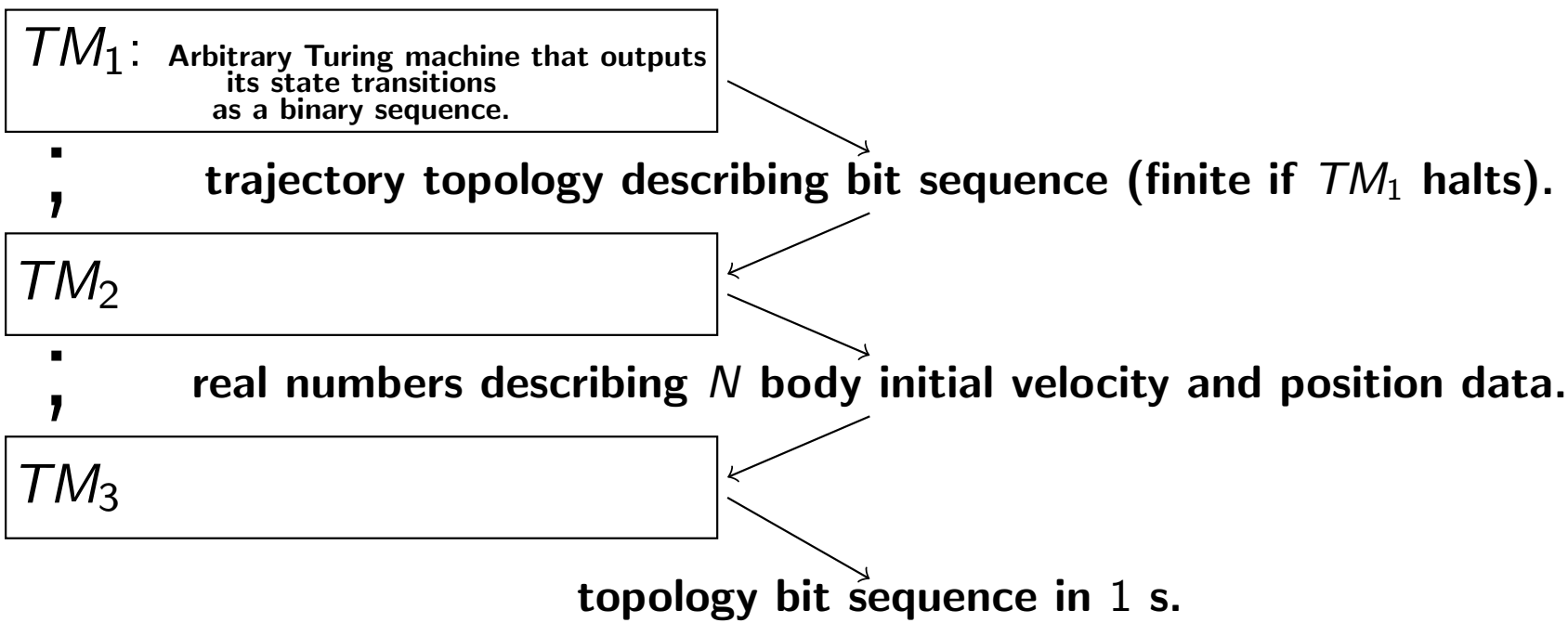


Topology



Singularity [Smith 2006]

TM:



Singularity [Smith 2006]

Description of TM_3 Given the initial real number data in such a form that it can access more bits on demand, by some integration scheme, simulates the motion of the n-body system to sufficient accuracy to be confident it knows the topology of the trajectories the bodies take in 1 s.

Theorem

TM_3 halts iff the N bodies do not reach the singularity in 1 s.

The Church-Turing Thesis

The Church-Turing Thesis

Postulate

Church-Turing thesis states that the set of things commonly understood to be computation is identical with the set of tasks that can be carried out by a Turing machine.

Postulate (As read by Kreisel 1987; statement 1)

Any physical system that purports to be a computer is not capable of any computational task that a Turing machine in incapable of.

The Church-Turing Thesis

Postulate

Church-Turing thesis states that the set of things commonly understood to be computation is identical with the set of tasks that can be carried out by a Turing machine.

Postulate (As read by Kreisel 1987; statement 1)

Any physical system that purports to be a computer is not capable of any computational task that a Turing machine in incapable of.

The Church-Turing Thesis

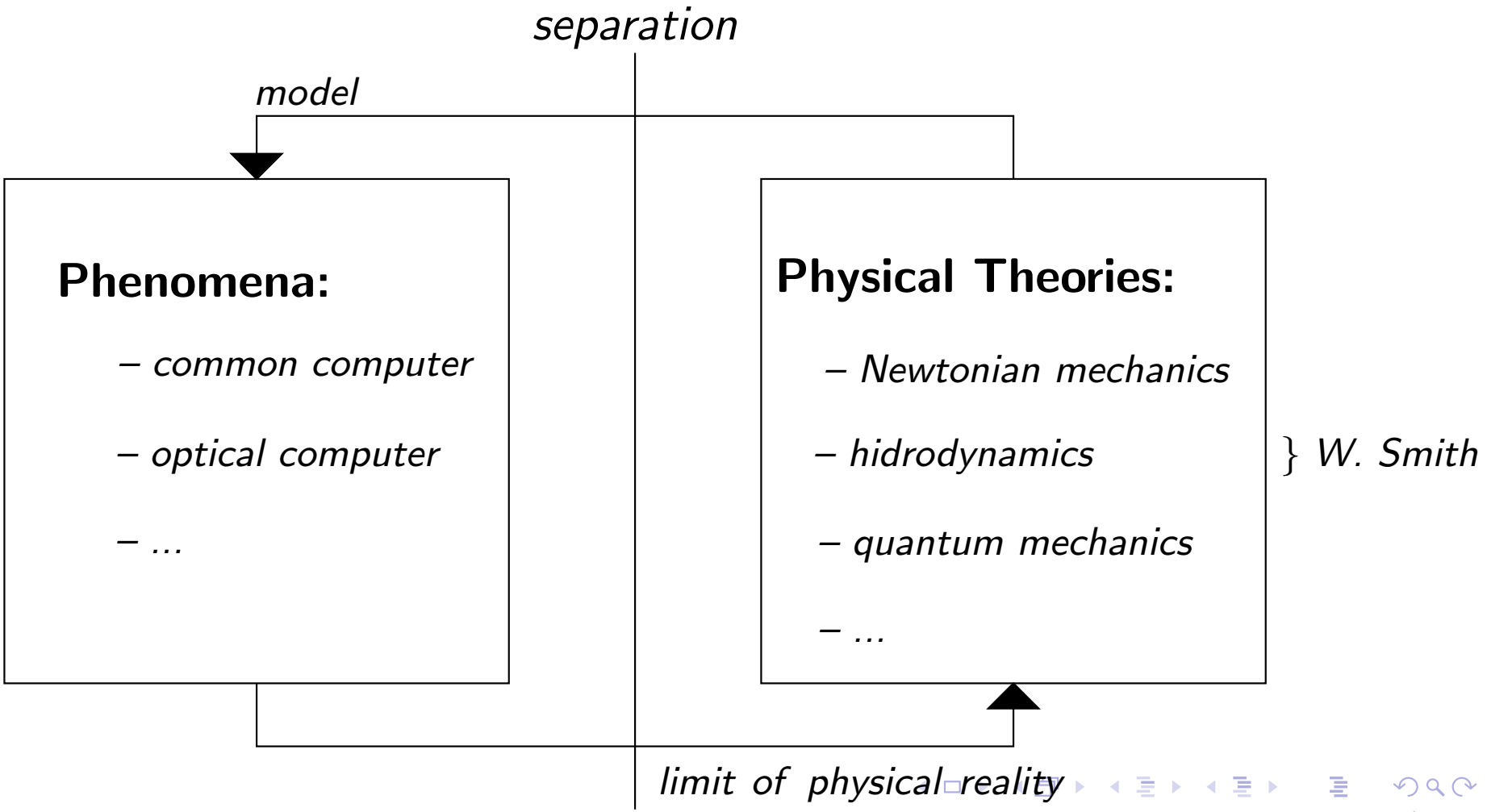
Postulate

Church-Turing thesis states that the set of things commonly understood to be computation is identical with the set of tasks that can be carried out by a Turing machine.

Postulate (As read by Kreisel 1987; statement 1)

Any physical system that purports to be a computer is not capable of any computational task that a Turing machine in incapable of.

Kreisel's perspective



Kreisel's perspective

Postulate (As read by Kreisel 1987; statement 2)

A theory is mechanistic if every sequence of natural numbers or every real number which is well defined (observable) according to theory is recursive or more generally, recursive in the data.

Postulate (The standard CT thesis)

Every function computable by an abstract human being following a routine procedure is Turing machine computable.

Every function computable by a finite mechanical procedure is computable by a Turing machine.

Postulate (The Physical CT thesis)

Every function computable by a finite physical system is Turing machine computable.

Postulate (The simulation thesis)

Every function simulating a finite physical system is Turing machine computable.

Postulate (The standard CT thesis)

Every function computable by an abstract human being following a routine procedure is Turing machine computable.
Every function computable by a finite mechanical procedure is computable by a Turing machine.

Postulate (The Physical CT thesis)

Every function computable by a finite physical system is Turing machine computable.

Postulate (The simulation thesis)

Every function simulating a finite physical system is Turing machine computable.

Postulate (The standard CT thesis)

Every function computable by an abstract human being following a routine procedure is Turing machine computable.
Every function computable by a finite mechanical procedure is computable by a Turing machine.

Postulate (The Physical CT thesis)

Every function computable by a finite physical system is Turing machine computable.

Postulate (The simulation thesis)

Every function simulating a finite physical system is Turing machine computable.

Postulate (The standard CT thesis)

Every function computable by an abstract human being following a routine procedure is Turing machine computable.
Every function computable by a finite mechanical procedure is computable by a Turing machine.

Postulate (The Physical CT thesis)

Every function computable by a finite physical system is Turing machine computable.

Postulate (The simulation thesis)

Every function simulating a finite physical system is Turing machine computable.

Church-Turing thesis

Abstract of Smith's paper on the n-body system

Church's thesis is at the foundation of computer science. We point out that any particular set of physical laws, Church's thesis need not merely be postulated, in fact it may be decidable. Trying to do so is valuable. In Newton's laws of physics with point masses, we outline a proof that Church's thesis is false; physics is unsimulable. But with certain more realistic laws of motion, incorporating some relativistic effects, the extended Church's thesis is true.

Church-Turing thesis

Philosophic question If Warren's proof had been done in the beginning of the XX century, would the physicists have tried to reformulate Newtonian physics to make Physics simulatable and reestablish the Church-Turing thesis?

Can we use a computational perspective (such like CT) as a refutation tool of a scientific theory? If not, what is the meaning of a non simulatable scientific theory?

Church-Turing thesis

Philosophic question If Warren's proof had been done in the beginning of the XX century, would the physicists have tried to reformulate Newtonian physics to make Physics simulatable and reestablish the Church-Turing thesis?

Can we use a computational perspective (such like CT) as a refutation tool of a scientific theory? If not, what is the meaning of a non simulatable scientific theory?

Church-Turing thesis

Philosophic question If Warren's proof had been done in the beginning of the XX century, would the physicists have tried to reformulate Newtonian physics to make Physics simulatable and reestablish the Church-Turing thesis?

Can we use a computational perspective (such like CT) as a refutation tool of a scientific theory? If not, what is the meaning of a non simulatable scientific theory?